AD-A051 840 MASSACHUSETTS UNIV AMHERST DEPT OF INDUSTRIAL ENGINE--ETC F/G 15/5
MANPOWER SCHEDULING MODELS IN SERVICE OPERATIONS,(U)
DEC 76 S CHONG, R J GIGLIO DAAG17-75-C-0017

UNCLASSIFIED NL

| OF |

AD
A051840

END
DATE
FILMED
5-78
DDC

(1)

(6) # Manpower Scheduling Models In Service Operations

by

(10) Sing-Choi /Chong

and

Richard J./Giglio

(15) Prepared under Contract Number DAAG17-75-C-Ø017
for the US Army Natick Laboratories
(11) December, 1976 (12) 65p.

partment of Industrial Engineering and Operations Research .
University of Massachusetts Univ., Amherst.
Amherst, Massachusetts

S/C

390 Ø85 — [illegible]

TABLE OF CONTENTS

Chapter 1

Introduction

1.1. Problem Statement

Manpower Scheduling is the process of converting daily or hourly workforce requirements into precise scheduling assignments by specifying both the days and the shift patterns each employee works. Manpower scheduling is important in all business operations, and is especially critical in the case of service operations because services often cannot be inventoried or deferred, and because minimum manpower requirements vary from hour to hour, shift to shift, and day to day. Examples of service operations are telephone directory assistance, turnpike toll collection, airline reservation booking, and food service operations -- the application which prompted this research.

The scheduling of workers has become more important lately because of the rapid increase in labor costs in both government and private firms. One recent study by Smith, et al [29], showed that labor costs in one typical large military installation amounted to nearly $12 million or 55% of the total annual cost of food service operations. Also, in another study, Lunberg and Armatas [4] stated that in restaurant operations, as much as 33% of the available labor is wasted through lack of proper scheduling.

Ahuja and Sheppard [2] reported that a computerized nurse scheduling system utilized in General Hospital, St. Johns, Newfoundland was able to reduce nurse overtime and simplify adjusting for vacation and sick leave.

Mapstone and Thamaro [22] also reported that significant savings were obtained when a vacation manpower scheduling model was used to schedule employees' vacation so as to spread vacations as evenly as possible over vacation planning period.

The idea of using an automated (computerized) manpower scheduling system is not new. However, the materialization of such a system is slow because manpower scheduling problems are complex, and the development of innovative mathematical models and algorithms is still in its infant stage. Undoubtedly, with increase interest generated by discoveries on the part of Operations Research scientists and greater awareness of managers in the advantages of using an automated system, manpower scheduling will rank as one of the most promising candidates for improving operations efficiency.

1.2 Research Objective

This research arises as a result of a contract with the U.S. Army Natick Labs to study and develop manpower scheduling algorithms which would efficiently schedule food service workers at military installations.

The objective of this research is to provide a manager with an automated (computerized) program for scheduling manpower. At the present time, the scheduling of food service workers at a military installation is accomplished manually by each of the dining hall stewards. The number of workers to be scheduled at a given dining facility ranges from 12 to 125 depending upon the number of customers and hours of operation.

Some of the benefits to be realized by implementation of an automated scheduling system are expected to include:

(a) saving of time spent by the manager to create a schedule,

(b) saving in labor cost due to a more efficient schedule,

(c) improved employee morale due to a fairer and more efficient schedule, and

(d) better customer service because employees are working in the right place at the right time.

Chapter 2

Literature Review

Manpower Scheduling Overview

Manpower Scheduling can be viewed as a subset of Manpower Planning, which includes manpower forecasting, selection and placement of personnel, production and maintenance of human capacities [18]. Manpower Scheduling can also be viewed as a subset of Scheduling which includes scheduling machines or jobs in a job shop environment [12], the scheduling of capital funds or cash in and outflows in a business enterprise [25], or the scheduling of materials and parts in a production plant [15]. A great amount of Literature including texts have been devoted to the topics of Manpower Planning and Scheduling e.g. [7,8, 24,25]. This review here will confine itself to the topic of manpower scheduling.

Manpower Scheduling problems are not confined to service operations and arise in many different environments and situations. For the purpose of discussion it is convenient to divide manpower scheduling problems into the following categories:

(a) Assignment Problems - In many business organizations, there often arises the problem of assigning workers to jobs so as to minimize the total cost/maximize the total efficiencies, given that the cost/efficiency of each worker is different when assigned to different jobs. Efficient network algorithms have been used successfully in solving the above problem. [10]

(b) Manufacturing - Many manpower scheduling problems exist in Manufacturing Operations. One of the more well known problems is the balancing of the production/assembly line, where workers are assigned to different work stations in order to create a balanced production line. [15]

(c) Airlines - The flight crew scheduling problem can be described as follows. Given the airline's timetable, a large set of possible crew rotations can be generated. Each crew rotation is a segment of scheduled flight segments constituting a round trip - that is, a sequence departing from and returning to one of the airline's crew bases. Each rotation must comply with all of the relevant federal, company, and union regulations. The problem then is to select an optimal subset of all flyable rotations. [3]

(d) Hospitals - The principal work carried out in this area concerned nurse scheduling in hospitals and physician assignments in out-patient clinics. In the case of physicians scheduling, the problem concerns the allocation of available resources in a manner which accomodates patient demands while retaining a high degree of physician utilization and satisfaction. [16] In nurse staffing, the problem concerns the efficient matching of workers while service demands are placed upon multiple work centers. The problem involves decisions relating to the basic organization and design of work centers, the interrelations among the type of staff at each center, the operation and control of the staffing process, and the training of nurses, and the short term scheduling of available staff to work centers. [1]

(e) Service Operations - Since this is the subject of current research, problems relating to this area will be described in the next section.

2.2 Review of Manpower Scheduling in Service Operations

There are several types of manpower scheduling problems. The literature reveals three general types of manpower scheduling problems in service operations. Problem type I, [5,17,23,26,30] involves allocating employees to fluctuating daily manpower requirements such that each employee has a certain days-off pattern, eg., two consecutive days off each week. A typical model is shown below:

$$\begin{array}{llllllll}
\text{Min} & \sum_{j=1}^{7} x_j & & & & & & \\
\text{st} & x_1 & & & + x_4 + x_5 + x_6 + x_7 & \geq & R_1 & \\
& x_1 + x_2 & & & + x_5 + x_6 + x_7 & \geq & R_2 & \\
& x_1 + x_2 + x_3 & & & + x_6 + x_7 & \geq & R_3 & \\
& x_1 + x_2 + x_3 + x_4 & & & + x_7 & \geq & R_4 & \qquad (1) \\
& x_1 + x_2 + x_3 + x_4 + x_5 & & & & \geq & R_5 & \\
& \quad x_2 + x_3 + x_4 + x_5 + x_6 & & & & \geq & R_6 & \\
& \qquad x_3 + x_4 + x_5 + x_6 + x_7 & & & & \geq & R_7 &
\end{array}$$

where x_j = number of employees with days-off pattern j.

$x_j \geq 0$ and integer, $j = 1,\ldots,7$.

R_i = minimum number of employees required for day i.

Tibrawala, et al, [30] and Baker [5] have both developed efficient algorithms which solve (1), without relying on Integer Programming. However, their algorithms fail to work for more complicated problems such as Problem type II and III described below.

Problem type II [19,27] concerns the allocation of employees with different shift patterns (that is, different patterns of working hours) to meet manpower requirements which change throughout a working day. Essentially, the mathematical formulation is similar to problem type I, except that shift patterns are generally quite different from days-off patterns. A typical model is shown below:

$$\begin{aligned} \text{Min} \quad & \sum_j c_j x_j \\ \text{st} \quad & a_{ij} x_j \geq R_i \qquad i = 1,\ldots,m \\ & x_j \geq 0 \text{ and integer} \end{aligned} \tag{2}$$

where a_{ij} = 1 if shift pattern j requires an employee to be working during period i.
0 otherwise.

m = number of periods (eg., hours) in a working day.

R_i = minimum manpower requirements for ith period.

X_j = number of employees in shift pattern j.

C_j = cost of one employee in shift pattern j.

Problems where the shift patterns consist of only contiguous periods, can be formulated as a transshipment problem [10]. However, for problems involving shift patterns which are split shifts (i.e., contiguous working periods followed by a rest period and then followed by contiguous working periods), an integer program approach is required unless the number of workers is so large that fractional answers can be rounded up or down with no appreciable loss of accuracy.

Segal [27] used a transshipment algorithm as a first approximation to his problem of scheduling telephone operators to ideal shift patterns (which are split shifts) so as to satisfy minimum quarter-hourly manpower requirements throughout the working day. Fitting his approximate solution to the ideal shift patterns, he then moved excess workers to deficient periods. This he accomplished by means of another network formulation. His method is reported to provide good, but not necessarily the optimal, schedules.

Problem type III is essentially a combination of Problem type I and II. Here the task is to allocate employees to different shift and days-off patterns such that the fluctuating manpower requirement for each period of each day of the week is satisfied. For any realistic problem, the integer programming formulation for Problem type III is usually very large, so individuals such as Smith [28] and Luce [21] have developed heuristic algorithms which give good but not necessarily optimal solutions.

Smith's algorithm [28], consists essentially of 2-phases. Phase 1 solves for the optimal manpower schedule for each day of the week without considering days-off pattern. In phase 2, the phase 1 solutions are joined up using the Tibrawala's algorithm. In joining up the solutions, several heuristic rules were followed. The solution obtained was reported to be good.

Chapter 3

Theory and Algorithm Development

3.1 Problem Type I

Problem Type I involves allocating employees to fluctuating daily manpower requirements such that each employee has a specified days-off pattern. As presented in Section 2.1,(1), involves days off patterns where each employee is given two consecutive days off each week. However, the formulation can easily be modified to handle the situation where the days off need not be consecutive. Since there are ${}^{7}C_{2}$ combinations of 2 days off out of 7 days in a week the number of variables increases to 21 rather than 7.

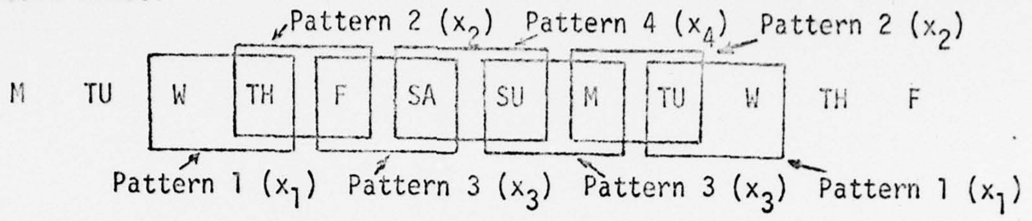
$$\begin{aligned} &\text{Min} \sum_{j=1}^{21} x_j \\ &\text{st} \sum_{j=1}^{21} A_j x_j \geq R \qquad (3) \\ &\quad x_j \geq 0 \text{ and integers} \end{aligned}$$

where $R = (R_1, R_2, \ldots R_7)^T$

$A_j = (a_{ij}, a_{2j}, \ldots a_{7j})^T \quad a_{ij} = 0 \text{ or } 1 \text{ and } \sum_{i=1}^{7} a_{ij} = 5 \text{ for } j=1\ldots, 21$

Very often, many business organizations have manpower requirements which are essentially the same for week days (M-F) but different for weekends (Sa-Su). (For example, food service operations in military installations). If employees are allowed two consecutive days off per week, (1) can then be simplified by reducing the number of variables to 4 days off patterns.

In the following figure assume that the manpower requirements for a week are symmetrical about the weekends. The days off paterns are identified by their distance from the weekends, and are represented by enclosed boxes.



The formulation becomes:

$$\text{Min} \sum_{j=1}^{4} x_j$$

$$\begin{aligned}
\text{st.} \quad && x_2 + x_3 + x_4 &\geq R_W \\
&& x_1 + x_2 + 2x_3 + 2x_4 &\geq R_{Tu} + T_{Th} \\
&& 2x_1 + x_2 + x_3 + 2x_4 &\geq R_M + R_F \qquad (4)\\
&& 2x_1 + 2x_2 + x_3 &\geq R_{Sa} + R_{Su} \\
&& X_j &\geq 0 \text{ and integer}
\end{aligned}$$

where X_j = Number of employees with days-off pattern j

R_i = Manpower requirements for day i

i = M, Tu, W, Th, F, Sa, Su

Similarly, if employees are allowed two days off per week, not necessarily consecutive, (2) can then be simplified by reducing the number of variables to 3 days-off patterns as follows.

let x_1 = Number of employees with pattern 1 (2 weekend days off)

x_2 = Number of employees with pattern 2 (1 weekend day and 1 weekday off)

x_3 = Number of employees with pattern 3 (2 weekdays off)

The formulation becomes:

$$\text{Min} \sum_{j=1}^{3} x_j$$

$$\text{st } 5x_1 + 4x_2 + 3x_3 \geq R_m + R_{tu} + R_w + R_{th} + R_f$$

$$x_2 + 2x_3 \geq R_{sa} + R_{su} \qquad (5)$$

Each of the problems (1), (2), (3), (4), and (5) can be solved by integer program algorithms.

3.2 Problem Type II

3.2.1 Problem Description

Problem Type II involved the allocation of employees with different shift patterns to period by period manpower requirements throughout a working day. (2) in Section 2.2 is a typical model. It was noted there that for problems involving shift patterns which are split shifts, an integer program code is necessary to solve the problem whereas, for problems involving shift patterns consisting of contiguous periods, the problem can be formulated as a transshipment problem.

Now, consider the following problem involving both contiguous and split shift patterns.

Let C_j = cost of one employee working contiguous shift pattern j

f_j = cost of one employee working split shift pattern j

x_j = number of employees working contiguous shift pattern j

y_j = number of employees working split shift pattern j

s_k = slack variable for equation (period) k, $k = 1, 2, \ldots,$ number of periods

R_k = manpower requirements for period k.

The formulation of Problem Type II can be written as

$$\text{Min} \sum_j C_j x_j + \sum_j f_j y_j$$

$$\begin{array}{cccccc|ccc|cccccccccc}
x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & y_1 & y_2 & y_3 & s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & & & \\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & = & R_1 & \text{(i)} \\
1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & = & R_2 & \text{(ii)} \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & = & R_3 & \text{(iii)} \\
1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & = & R_4 & \text{(iv)} \\
0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & = & R_5 & \text{(v)} \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & = & R_6 & \text{(vi)} \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & = & R_7 & \text{(vii)}
\end{array} \qquad (6a)$$

If we subtract equation (j-i) from (j), for j=(ii) to (vii) and if we substract $\vec{0}$ from equation (i) and equation (vii) from $\vec{0}$, then we obtain

$$\begin{array}{cccccc|ccc|ccccccccc}
1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & = & R_1 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & -1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & = & R_2 - R_1 \\
0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & = & R_3 - R_2 \\
0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & = & R_4 - R_3 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & = & R_5 - R_4 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & = & R_6 - R_5 \\
0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & = & R_7 - R_6 \\
0 & 0 & 0 & -1 & 0 & -1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & = & -R_7
\end{array} \qquad (6b)$$

Defining matrices as expressed by the partitions above, (6b) can be written as

$$(A^1|F|A^2)\begin{pmatrix} X \\ \overline{Y} \\ \overline{S}\end{pmatrix} = \hat{R} \quad \text{or} \quad (A^1|A^2)\left(\frac{X}{S}\right) + FY = \hat{R}$$

Notice that every column in $(A^1|A^2)$ has only one positive 1 and one negative 1. This suggests that if $F = [0]$, $(A^1|A^2)\left(\frac{X}{S}\right) = \hat{R}$ is a transshipment problem. Notice also that $\sum_{i=1}^{8} \hat{R}_i = 0$, which suggests that the supply is equal to the demand in the transshipment problem, provided $(A^1|A^2)\left(\frac{X}{S}\right) = \hat{R}$ is consistent. One can represent the transshipment problem as a network by dividing $\hat{R}_i$ into 3 sets $G-$, G_0, G_+:

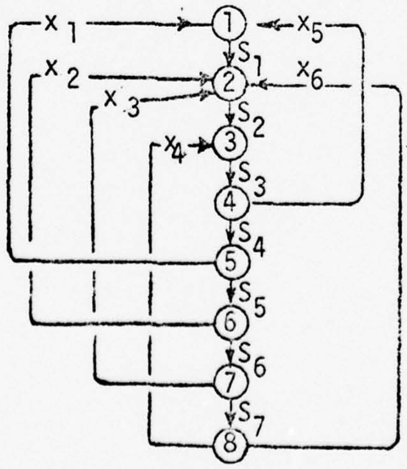
$$G- = \{\hat{R}_i < 0\}$$

$$G_0 = \{\hat{R}_i = 0\}$$

$$G_+ = \{\hat{R}_i > 0\}.$$

Using transshipment problem terminology $G-$ is associated with demand nodes.
G_0 is associated with intermediate stocking houses nodes.
G_+ is associated with supplying nodes.

The transshipment problem can be represented by the following network:



Where node i equals the beginning of working period i.

When $F \neq [0]$, we have the following problem:

$$\text{Min } CX + fY$$

$$\text{st. } (A^1|A^2)\left(\frac{X}{S}\right) + FY = \hat{R} \qquad (6c)$$

$$X \geq 0 \quad Y \geq 0 \text{ and integer vectors}.$$

The original problem (6a) has been partitioned into 2 sets. The first set of variables $(X|S)$ are those associated with the transshipment problem, and the second set (Y) with an integer program. The literature [11,13] reports successful attempts in solving problems having the above structure by means of the Benders Partitioning Method [6]. Using Bender's terminology, when Y is fixed, the linear sub-problem (transshipment problem) becomes:

$$\text{Min } CX + f\overline{Y}$$

$$\text{st. } (A^1|A^2)\left(\frac{X}{S}\right) = (\overline{R}-F\overline{Y}) = (\hat{\overline{R}}_1,\ldots,\hat{\overline{R}}_8)^T \qquad (6d)$$

Notice that the columns in F consist of equal numbers of positive and negative 1's. Therefore $\sum_{i=1}^{8} \hat{\overline{R}}_i = 0$, i.e. the supply is equal to the demand in the linear sub-problem, provided that the linear sub-problem is consistent.

3.2.2 Motivation and Background

Bender's algorithm utilized to solve (6a) evolved from the work of [6] and [10] and is founded on the naturally appealing idea of fixing $y = \bar{y} \in G$, solve a sub-problem to generate a $\bar{u}$, fix $u = \bar{u}$ and then solve another problem to obtain a new $\bar{y}$. Continuation of this type of solution technique allows us to solve a series of manageable sub-problems which yield feasible solutions to the overall system. As is true with many partitioning techniques, duality results are instrumental in guaranteeing convergence to optimality as well as providing upper and lower bounds on the objective value--the bounds being very informative when the procedure is terminated before the optimal solution is attained.

Before a step-by-step statement of the algorithm is presented, let us inspect the original problem (6a), called it Problem "P".

$$\begin{array}{ll} \text{Min} & cx + fy \\ \text{s.t.} & Ax + Fy \geq R \\ & x \geq 0 \quad y \in G \text{ where } G = \{\text{set of integers}\} \end{array}$$

On fixing $y = \bar{y} \in G$, we are left with the following linear sub-problem, $P(\bar{y})$, (transshipment problem).

$$\begin{array}{ll} \text{Min} & \theta(x, \bar{y}) = cx + f\bar{y} \\ \text{st} & Ax \geq R - F\bar{y} \\ & x \geq 0 \end{array}$$

The dual to the linear sub-problem $D(\bar{y})$ is:

$$\begin{array}{ll} \text{Max} & \phi(\bar{y}, u) = (b - F\bar{y})\, u + f\bar{y} \\ \text{st} & uA \leq c \\ & u \geq 0 \end{array}$$

An important property of problem $D(\bar{y})$ is that its solution space is independent of $\bar{y}$. It will be seen shortly that this allows us to successively generate $\bar{y}$'s without duplication and to obtain a lower bound on the objective value of "P".

Let $(\hat{x},\hat{y})$ be an optimal solution to "P". From the theory of duality [19], there exists an optimal solution $\hat{u}$ to $D(\hat{y})$ such that $\phi(y,u) = \theta(x,y)$. Furthermore, if we assume that $\bar{y}$ is such that $P(\bar{y})$ is consistent, $\theta(\bar{x},\bar{y}) = \phi(\bar{y},u) \geq \phi(\hat{y},\hat{u}) \geq \phi(\hat{y},\bar{u})$ where $\bar{x}$ and $\bar{u}$ are optimal solutions to $P(\bar{y})$ and $D(\bar{y})$ respectively. Thus, by fixing $y=\bar{y}\varepsilon G$ and solving $P(\bar{y})$, or $D(\bar{y})$, an upper bound on the optimal objective value is obtained, and, since $\bar{u}$ is feasible for $D(y)$ regardless of the value of y, by minimizing $\phi(y,\bar{u})$ over $y\varepsilon G$, a lower bound is obtained. These observations lay the foundation for an algorithm to solve "P". However, the iterative process which appears to be forming breaks down when $P(\bar{y})$ is inconsistent. In most other applications of Bender's algorithm, a well-known theorem of alternative known as Farkas' Lemma [19] is used to eliminate the y's which fall into this category. However, in the case of the present application, $P(\bar{y})$, being a transshipment problem, is inconsistent only when a non-existent route appears in the optimal solution. Based on most network algorithms, this will be reflected as a high cost in the objective function of $P(\bar{y})$, and assuming that the original problem "P" is consistent, the inconsistency in $P(\bar{y})$ will resolve itself as the algorithm proceeds.

3.2.3 Statement of Algorithm

An iterative procedure based on the preceding results is given below:

Step 1. Initialize the following. Set $\bar{y} = 0$, the upper and lower bounds on the optimal objective value of "P", $UB=+\infty$ and $LB=-\infty$. The number of linear sub-problems solved with a finite optimal solution, ks=0.

Step 2. Solve $P(\bar{y})$ and obtain the optimal dual variables $\bar{u}$.

Step 3. If $\theta(\bar{x},\bar{y}) < UB$, let $UB = \theta(\bar{x},\bar{y})$. Let ks = ks + 1 and $u^{ks} = \bar{u}$.

Step 4. Solve the following <u>master problem</u> (MP)

Minimize z

st $z \leq \phi(y,u^i)$, $j = 1,2,\ldots, ks$

$y \in G$

If $ks=0$, $\bar{z} = -\infty$ and constraint set of (MP) is omitted.

Step 5. Let $(\bar{z},\bar{y})$ be one optimal solution to (MP). Put $LB = \bar{z}$. If $LB = UB$, terminate the algorithm, otherwise go to step 2.

3.3 Problem Type III

3.3.1 Problem Description

Problem Type III is essentially a combination of problem type I and II. Here the problem is to allocate employees to different shift and days-off patterns such that the fluctuating manpower requirement for each period of each day of the week is satisfied. Several versions of the scheduling problem are described below.

Case III-A

(A1) The objective is to minimize the cost of employees.

(A2) The hour-by-hour manpower requirements for each day of the week cycle must be met.

(A3) Each employee works five consecutive days and is given two consecutive days off per week.

(A4) An employee can work one shift one day and another shift on another day.

(A5) The number of hours in each working day is no longer than sixteen hours, so that every employee is given at least eight hours of break before working again.

(A6) A full time shift is a shift of eight consecutive hours work.

(A7) A part time shift is a shift of four consecutive hours work.

(A8) A full time employee is one who works only in full time shifts. Similarly, a part time employee is one who works only in part time shifts.

(A9) Except for the last seven hours of any working day, there exists one full time shift beginning at every hour. Similarly, except for the last three hours of any working day, there exists one part time shift beginning at every hour.

(A10) The cost of an employee depends on whether s/he is a full time or a part time employee and does not depend on the shifts s/he is assigned to on each day. The cost of each full time employee is the same and the cost of each part time employee is also the same. The cost of one full time employee can be greater or less than the cost of two part time employees.

(A11) Each hour has at least one full time employee on duty.

Relaxations or modifications to the above conditions will be discussed in Section 3.3.4.

Case III-B

Same as Case III-A except that weekdays manpower requirement is the same but different from weekend requirements.

Case III-C

Same as Case III-A except that each employee is given 2 days off, not necessarily consecutive, each week.

Case III-D

Same as Case III-C except that the weekday manpower requirement is the same but different from weekend requirements.

Case III-E

Same as Case III-A except that a part time shift is 5 consecutive hours work.

The integer program formulation of the problem based on Case III-A conditions is first presented. Two formulations are given. A new 2-phase algorithm is then presented based on all 5 cases. Lastly, extension of the 2-phase algorithm to multiple shift patterns is presented.

3.3.1.1. Formulation I

$$\text{Minimize} \quad \sum_{j=1}^{q^F} C^F S_j^F + \sum_{j=1}^{q^P} C^P S_j^P$$

Subject to

$$\text{(IP1-a)} \quad \sum_{j=1}^{q^f} \delta_{ijk}^F S_j^F + \sum_{j=1}^{q^P} \delta_{ijk}^P S_j^P \geq R_{ik} \quad \text{for } i=1,\dots,7,\; k=1,\dots,m_i \qquad \text{(IP1)}$$

$$\text{(IP1-b)} \quad \sum_{j\varepsilon J_{i1}} S_j^F \geq 1 \text{ and } \sum_{j\varepsilon J_{iq^F}} S_j^F \geq 1 \text{ for } i=1,\dots,7$$

$$\text{(IP1-c)} \quad S_j^F \geq 0, \quad S_j^P \geq 0 \text{ and integers for all } j.$$

where,

C^F = cost of one fulltime (FT) employee.

S_j^F = number of FT employees working in shift/days-off pattern j.

Q^F = number of FT shift/days-off patterns.

δ_{ijk}^F = 1 if FT employee working in shift/days-off pattern j works during period k on day i.

0 otherwise.

C^P = cost of one part time (PT) employee

S_j^P = number of PT employees working in shift/days-off pattern j.

Q^P = number of PT shift/days-off patterns

δ_{ijk}^P = 1 if PT employee working in shift/days-off pattern j works during period k on day i.

0 otherwise

R_{ik} = minimum number of employees required during period k on day i

m_i = number of periods in working day i

J_{i1} = {shift/days off pattern j | employee works on the first full time shift pattern of day i}

J_{iq^F} = {shift/days off pattern j | employee works on the last full time shift pattern of day i}

Constraint set (IP1-a) ensures that the minimum requirements are satisfied.
Constraint set (IP1-b) ensures that on any period of the day, there is at least one full time employee on duty.
Constraint set (IP1-c) ensures that all variables are non-negative and integers.

For a typical problem where consecutive days-off patterns are assumed,

$$mi = 16$$

$$Q^F = n\,x(mi-7)^5 = 7 \times (9)^5 = 413,343$$

$$Q^P = n\,x(mi-3)^5 = 7 \times (13)^5 = 2,599,051$$

giving 3,012,394 variables and 126 constraints. Notice that the number of variables is huge.

3.3.1.2. Formulation II

$$\text{Minimize } C^F \sum_{g=1}^{1} X_g^F + C^P \sum_{g=1}^{1} X_g^P$$

Subject to:

$$\text{(IP2-a)} \quad \sum_{j=1}^{q^F} \delta_{ijk}^F S_{ij}^F + \sum_{j=1}^{q^P} \delta_{ijk}^P S_{ij}^P \geq R_{ik} \qquad \begin{array}{l} i = 1,\dots,7 \\ k = 1,\dots,m_i \end{array}$$

$$\text{(IP2-b)} \quad S_{i1}^F \geq 1$$

$$i = 1,\dots,7$$

$$S_{iq^F}^F \geq 1 \qquad \text{(IP2)}$$

(IP2-c) $$\sum_{g=1}^{7} \rho_{ig}^{F} X_g^{F} = \sum_{j=1}^{q^F} S_{ij}^{F}$$

$$\sum_{g=1}^{7} \rho_{ig}^{P} X_g^{P} = \sum_{j=1}^{q^F} S_{ij}^{P} \qquad i = 1,\ldots,7$$

(IP2-d) $S_{ij}^{F}, S_{ij}^{P}, X_g^{F}, X_g^{P} \geq 0$ and integer for all i, j, g

q^F = number of fulltime shift patterns

q^P = number of part time shift patterns

X_g^F = number of full time employees having days off pattern g

ρ_{ig}^F = 1 if full time employee with days off pattern g works on day i

0 otherwise

ρ_{ig}^P = 1 if part time employee with days off pattern g works on day i

0 otherwise

C^F = cost of a full time employee

C^P = cost of a part time employee

S_{ij}^F = number of full time employees working in shift j on day i

S_{ij}^P = number of part time employees working in shift j on day i

R_{ik} = minimum number of employees required on hour k of day i

M_i = number of periods in working day i

δ_{ijk}^F = 1 if full time employee works during period k in working shift pattern j on day i

0 otherwise

δ^P_{ijk} = 1 if part time employee works during period k in working shift pattern j on day i

0 otherwise

Constraint set (IP2-a) ensures that the minimum manpower requirements are satisfied.

Constraint set (IP2-b) ensures that on any period of any day, there is at least one full time employee on duty.

Constraint set (IP2-c) distributes the shift patterns of the seven days to different days off patterns.

Constraint set (IP2-d) ensures that all variables are non-negative and integers.

For the same problem as discussed in Formulation I, where $m_i = 16$, $q^F = 9$, $q^P = 13$, there are 140 constraints and 168 variables.

If we double the number of work periods (say each period = 1/2 hour) then $m_i = 32$, $q^F = 18$, $q^H = 26$ yielding 252 constraints and 322 variables.

Integer program codes suitable for handling problems of these magnitudes, although available in the market, are nevertheless costly to purchase as well as to run them. Also, large computers are necessary to run such large problems. Since most organizations have small computers, and attempt is made here to decompose the problem into several smaller problems. The next section proposes an algorithm to solve the problem by decomposing the problem into two phases, each phase consisting of smaller integer programs.

3.3.2 2-Phase Algorithm Based on Case III-A Conditions

In this section, a procedure is developed to solve Problem type III based on Case III-A conditions. The other cases will be presented in subsequent sections. The procedure involves 2 phases. Phase 1 consists of solving for the optimal schedule for each day of the week. Phase 2 will then connect up the optimal solutions obtained from Phase I by allowing days off for each employee. The motivation behind this 2 phase procedure is the fact that 2 consecutive part time (PT) shifts can replace 1 full time (FT) shift and vice versa. Also 1 full time shift can replace 1 PT shift if necessary.

Phase 1:

In Phase 1, the optimal solution for every day of the week is obtained using the integer program below, (one problem for each day, 7 problems altogether). In the first phase, a solution is desired such that if two consecutive PT shifts appear in the optimal solution, they will appear as one corresponding FT shift. This can be done by costing PT employees slightly more than 1/2 the cost of a FT employee, i.e., ($C^P = 1/2C^F + \varepsilon$) where ε is an arbitrarily small number. By moving all consecutive PT shifts to their corresponding FT shift in Phase 1, it becomes only necessary to consider replacing one FT shift with two consecutive PT shifts in Phase 2. The Phase 1 formulation for day i is similar to (2).

$$\text{Minimize } C^F \sum_{f=1}^{q^F} S_f^F + C^P \sum_{g=1}^{q^P} S_g^P \qquad (7)$$

$$\text{Subject to (7a)} \sum_{f=1}^{q^F} \delta_{kf}^F S_f^F + \sum_{g=1}^{q^P} \delta_{kg}^P S_g^P \geq R_k \text{ for } k = 1,\ldots m_i$$

(7b) $S_1^F \geq 1$

$S_{q^F}^F \geq 1$

(7c) $S_f^F \geq 0$ and integers for $f = 1,\ldots q^F$

$S_g^P \geq 0$ and integers for $f = 1,\ldots q^P$

Where C^F = cost of 1 FT employee.

S_f^F = number of FT employee in FT shift f.

C^P = cost of 1 PT employee.

S_g^P = number of PT employee in PT shift g.

q^F = number of FT shift patterns.

q^P = number of PT shift patterns.

δ_{kf}^F = 1 if FT employee works during period k in FT shift pattern f. 0 otherwise.

δ_{kg}^F = 1 if PT employee works during period k in PT shift pattern g. 0 otherwise.

R_k = Minimum number of employees required during period k.

m_i = number of periods in working day i.

Constraint Set (7a) ensures that the minimum manpower requirements for each hour are satisfied.

Constraint Set (7b) ensures that during any period of the day, there is at least one full time employee on duty.

Constraint Set (7c) ensures that all variables are non-negative and integers. Since the shift patterns are contiguous, problem (7) can be formulated as a transshipment problem discussed in Section 3.2. If the shift patterns were not contiguous, (7) would have to be solved using integer programming. Let us assume that the solution obtained is as follows:

$$\sum_{f=1}^{q^F} S_f^{F*} = \alpha_i$$

$$\sum_{g=1}^{q^P} S_g^{P*} = \beta_i$$

Since the length of a FT shift is twice that of a PT shift,

$$\text{Solution} = \alpha_i \text{ FT} + \beta_i \text{ PT}$$
$$= (2\alpha_i + \beta_i) \text{ PT equivalents}$$

We know that this is the best (i.e., least cost) schedule we can obtain for each particular day i.

Phase 2:

In Phase 2, the seven optimal solutions for each day of the week are connected by another integer program. In this formulation, we have to make sure that each employee is given two consecutive days off per week. Also, two replacement possibilities have to be represented in the formulation: replacing one FT with two consecutive PT shifts and replacing one PT by one FT shift. The other possibility, that of replacing two consecutive PT shifts by one FT shift do not need to be considered because by costing PT employees slightly more than 1/2 the cost of a FT employee in Phase 1, all consecutive PT shifts are moved to their corresponding FT shift.

Minimize $C^F \sum_{j=1}^{7} x_j + C^P \sum_{j=1}^{7} y_j$

Subject to:

$$\begin{array}{lllllll}
 & x_1 & x_4 + x_5 + x_6 + x_7 & -F_1 & -Q_1 & = 0 & \\
 & x_1 + x_2 & x_5 + x_6 + x_7 & -F_2 & -Q_2 & = 0 & \\
 & x_1 + x_2 + x_3 & x_6 + x_7 & -F_3 & -Q_3 & = 0 & \\
\text{(8a)} & x_1 + x_2 + x_3 + x_4 & x_7 & -F_4 & -Q_4 & = 0 & \\
 & x_1 + x_2 + x_3 + x_4 + x_5 & & -F_5 & -Q_5 & = 0 & \\
 & \quad x_2 + x_3 + x_4 + x_5 + x_6 & & -F_6 & -Q_6 & = 0 & \\
 & \qquad x_3 + x_4 + x_5 + x_6 + x_7 & & -F_7 & -Q_7 & = 0 & \\
 & & & & & & \text{(8)} \\
 & y_1 & y_4 + y_5 + y_6 + y_7 & -P_1 & +Q_1 & = 0 & \\
 & y_1 + y_2 & y_5 + y_6 + y_7 & -P_2 & +Q_2 & = 0 & \\
 & y_1 + y_2 + y_3 & y_6 + y_7 & -P_3 & +Q_3 & = 0 & \\
\text{(8b)} & y_1 + y_2 + y_3 + y_4 & y_7 & -P_4 & +Q_4 & = 0 & \\
 & y_1 + y_2 + y_3 + y_4 + y_5 & & -P_5 & +Q_5 & = 0 & \\
 & \quad y_2 + y_3 + y_4 + y_5 + y_6 & & -P_6 & +Q_6 & = 0 & \\
 & \qquad y_3 + y_4 + y_5 + y_6 + y_7 & & -P_7 & +Q_7 & = 0 &
\end{array}$$

(8c) $2F_i + P_i \geq 2\alpha_i + \beta_i$ for $i = 1,\ldots 7$

(8d) $\gamma_i \leq F_i \leq \alpha_i$ for $i = 1,\ldots 7$

(8e) $x_j \geq 0$, $y_j \geq 0$, $F_i \geq 0$, $P_i \geq 0$, $Q_i \geq 0$ and integers for $i = 1,\ldots,7$ $j = 1,\ldots,7$

Where γi = minimum number of FT for any day i. For $m_i \leq 8$, $\gamma_i = 1$; for $8 < m_i \leq 16$; $\gamma_i = 2$ to ensure that a FT employee is working every hour of the day.

x_j = number of full-time employees whose days off pattern is j.

F_i = number of full-time employees required on day i, not counting those created from PT.

y_j = number of part-time employees whose days off pattern is j.

P_i = number of part-time employees required on day i, not counting those converted to FT.

Q_i = number of full time employees required on day i, who are created from equal number of PT (where 1 PT is replaced by 1 FT).

Constraint (8c) shows that $(2\alpha_i + \beta_i)$ PT equivalents is the minimum manpower which will meet the demand on the ith day.

Constraint (8a) and (8b) specify the days off patterns. Also, constraint (8b) allows a PT shift to become a FT shift and so is removed from (8b) and added on to (8a). This flexibility to allow a PT shift to become a FT shift is built in to allow some PT shifts to join up with FT shifts if that creates a better schedule. Constraint (8d) ensures that for any day i, the number of FT shifts must be greater than γ_i but not greater than α_i.

Rewriting (8) in a more convenient form yields

$$\text{Minimize } C^F \sum_{j=1}^{7} x_j + C^P \sum_{j=1}^{7} y_j$$

Subject to:

st (9a) $\sum_{j=1}^{7} A_j x_j - F - Q = 0$

(9b) $\sum_{j=1}^{7} A_j y_j - P + Q = 0$ (9)

(9c) $2F + P \geq 2\alpha + \beta$

(9d) $\gamma \leq F \leq \alpha$

(9e) $X \geq 0,\ Y \geq 0,\ F \geq 0,\ P \geq 0,\ Q \geq 0$ are integer vectors

and γ, β, α are all vectors corresponding to (8).

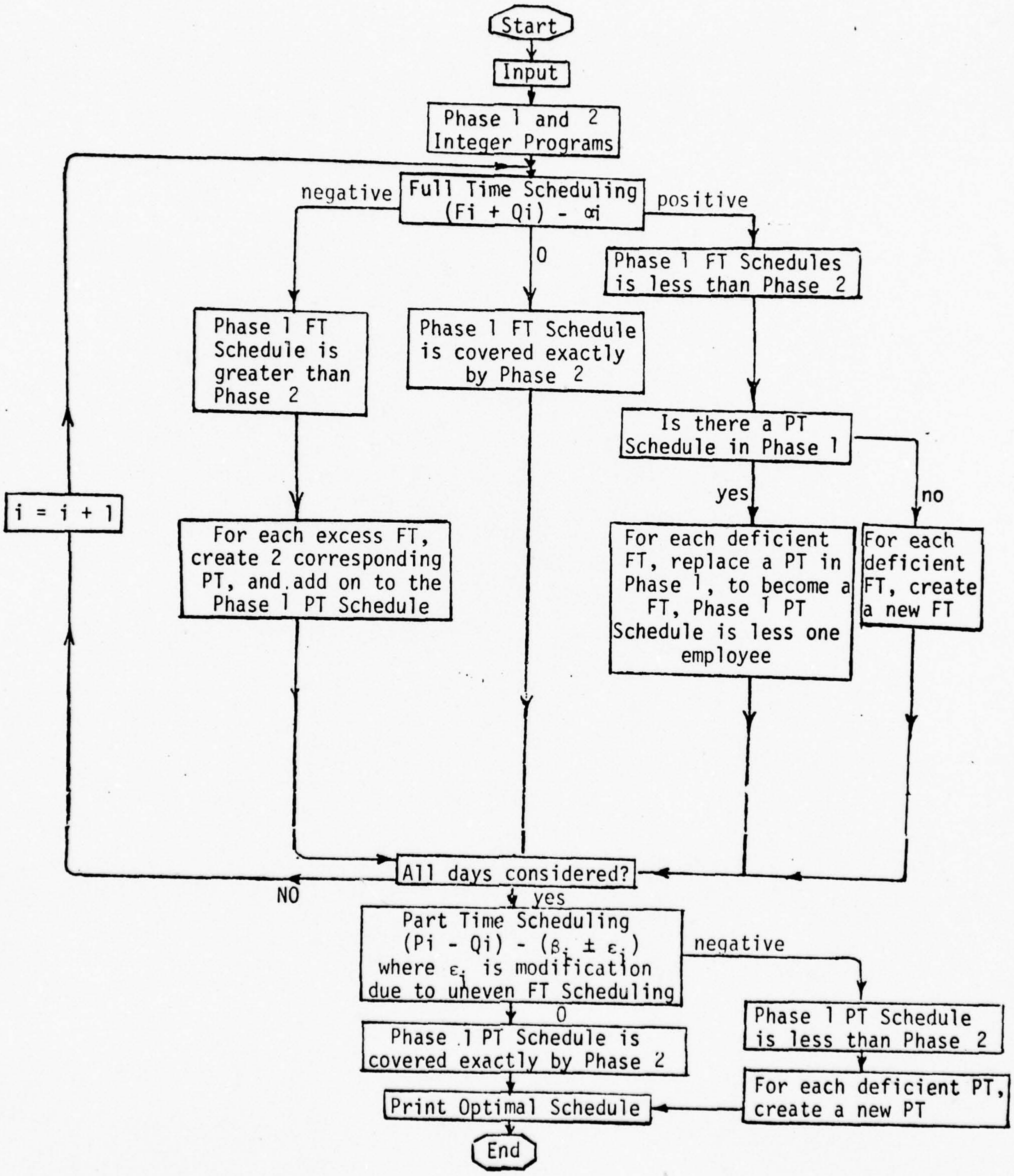
A_j is the column vector associated with the days off pattern j, and is consistent with the column vector used in (1).

Solution Schedule:

x_j's and y_j's are the optimal number of FT and PT employees. They will cover the daily requirements as indicated by the schedule obtained in phase 1. Obtaining actual manpower assignments is a straight-forward but cumbersome process which is best explained by a flow chart as shown in figure 1.

Figure 1

Macro - Flow Chart Showing How Final Optimal Schedule Is Created From Phase 1 and 2 Solutions



Start
Input
Phase 1 and 2 Integer Programs
Full Time Scheduling $(Fi + Qi) - \alpha i$
negative
0
positive
Phase 1 FT Schedule is greater than Phase 2
Phase 1 FT Schedule is covered exactly by Phase 2
Phase 1 FT Schedules is less than Phase 2
Is there a PT Schedule in Phase 1
yes
no
i = i + 1
For each excess FT, create 2 corresponding PT, and add on to the Phase 1 PT Schedule
For each deficient FT, replace a PT in Phase 1, to become a FT, Phase 1 PT Schedule is less one employee
For each deficient FT, create a new FT
All days considered?
NO
yes
Part Time Scheduling $(Pi - Qi) - (\beta_i \pm \varepsilon_i)$ where ε_i is modification due to uneven FT Scheduling
negative
0
Phase 1 PT Schedule is covered exactly by Phase 2
Phase 1 PT Schedule is less than Phase 2
For each deficient PT, create a new PT
Print Optimal Schedule
End

3.3.3 Problem Sizes

The 2-phase algorithm exploits the structure of the scheduling problem. For a problem involving 7 days, 16 working hours a day, Phase 1 consists of seven integer programs (Problem (7)), each having 22 variables and 18 constraints. Phase 2 consists of problem (8) which has 35 variables and 35 constraints for which 14 are upper/lower bounds on variables. Notice that all the problems are small, involving not more than 35 constraints. There are many advantages in having to solve seven much smaller integer programs rather than one huge one. Some of the advantages are:

(a) saving in computer memory storage

(b) saving in computer running times

(c) a small computer can be used .

3.3.4. Discussion of Case III-A Conditions

This section is devoted to discussing how the 2-phase algorithm is affected when the conditions of Case III-A are modified for type III problems.

Condition (A1)

The minimization of cost is usually acceptable to most organizations provided that certain constraints are met.

Condition (A2)

The hour-by-hour manpower requirements can be modified to half hourly or even quarter hourly requirements.

Condition (A3)

Any kind of days off patterns can be represented in the formulation of the Phase 2 problem (eg. see Section 3.3.6., Case III-C)

Condition (A4)

For service operations, the flexibility to allow an employee to work one shift on one day and another shift on another day is usually acceptable. When creating the final optimal Schedule from Phase 1 and 2 solutions, one can assign employees to shifts which are as close as possible on different days, but does not guarantee that the same shift is assigned on different days.

Condition (A5)

If the number of hours in each working day is greater than sixteen hours, then the condition that a break of at least eight hours before working again is not guaranteed. If the working day is twenty-four hours, a new problem type IV arises, as discussed in Section 3.4.

Condition (A6) and (A7)

Split shifts can be handled by the 2-phase algorithm.

A split-shift is any non-contiguous shift pattern. Since the number of split shifts is large and since many are not reasonable from an employee's point of view, several assumptions have to be made when developing an algorithm and programming a computer code.

- A split-shift has only one break of up to four hours
- Any full time shift (split or contiguous) which can possibly encompass a part-time split shift, and therefore, be a feasible replacement for it, is a valid full time shift.

When a full-time shift is not an integral multiple of a part time shift the 2-phase algorithm can be modified as shown in Section 3.3.8., Case III-E. When multiple shifts, i.e. three or more types of shifts of different lengths are involved, the 2-phase algorithm becomes complicated as shown in S⸱tion 3.3.9.

Condition (A8)

It is generally acceptable to have full time employees working in full time shifts only and part time employees working in part time shifts only.

Condition (A9)

If certain shift patterns are undesirable, one could cost these shift patterns many times more than the cost of the desirable shift patterns. However, the phase 2 problem is blind to the fact that certain shift patterns are undesirable, and so may produce an "optimal" schedule involving undesirable shifts if using them utilizes less manpower. Undesirable shift patterns will not be in the solution if full time shifts and their corresponding part time shifts are removed simultaneously. Also, the algorithm can search among existing shift patterns and choose only the desirable shift patterns to be included in the final schedule. This can happen in three instances:

(a) Phase 2 requires some of the FT shifts to become two PT shifts. We assume that all the FT shifts in Phase 1 solution are desireable shifts (this assumption is reasonable unless user specified too many shifts as undersirable, so that a feasible solution must consist of an undesirable shift). The algorithm has to search for those FT shifts in Phase 1 solution which can be broken up into two desirable PT shifts. Although unlikely, the search may not be successful, in which case, an undesirable shift has to appear in the final schedule. The user should be notified with an appropriate message.

(b) Phase 2 requires some of the PT shifts to become FT shifts. Again we assume that all the PT shifts in Phase 1 solution are desirable shifts. The algorithm has to search for a PT shift which can be replaced by a desirable FT shift. Although unlikely, the search may not be successful, in which case an undesirable shift has to appear in the final schedule. The user should be notified with an appropriate message.

(c) Phase 2 requires a FT or PT shift to be created. The algorithm has to choose among the desirable shift patterns.

Condition (A10)

The cost differential between shift patterns of the same length is usually small for most organizations.

Condition (A11)

The condition to have at least one full time employee on duty at each hour of the working day can be removed. Additional constraints like having a desired ratio of full time employees to part time employees can also be incorporated.

3.3.5 Case III-B Conditions

When the weekday requirements are essentially the same but different from weekend requirements, the 2-phase algorithm can be outlined as follows:

Phase 1:

Solve for optimal solutions to a weekday and a weekend day. (Two problems.) Let the solution be $(2\alpha^1 + \beta^1)$ PT equivalents and γ^1 for weekday, and $(2\alpha^2 + \beta^2)$ PT equivalents and γ^2 for weekend day.

Phase 2:

Consistent with the days off patterns as in (4), the Phase 2 formulation has fewer variables and constraints than (8) or (9). Using the compact form as in (9), we have,

$$\text{Minimize } C^F \sum_{j=1}^{4} x_j + C^P \sum_{j=1}^{4} y_j$$

$$\begin{aligned}
\text{st.}\quad & (10a) && \sum_{j=1}^{4} A_j x_j - F - Q = 0 \\
& (10b) && \sum_{j=1}^{4} A_j y_j - P + Q = 0 \\
& (10c) && 2F_1 + P_1 \geq 2\alpha^1 + \beta^1 \\
& && 2F_i + P_i \geq 2\,(2\alpha^1 + \beta^1) \text{ for } i = 2, 3 \\
& && 2F_4 + P_4 \geq 2\,(2\alpha^2 + \beta^2) \\
& (10d) && \gamma^1 \leq F_1 \leq \alpha^1 \\
& && 2\gamma^1 \leq F_i \leq 2\alpha^1 \text{ for } i = 2, 3 \\
& && 2\gamma^2 \leq F_4 \leq 2\alpha^2 \\
& (10e) && X \geq 0,\ Y \geq 0,\ F \geq 0,\ P \geq 0,\ Q \geq 0 \text{ and integer vectors.}
\end{aligned} \qquad (10)$$

where Aj is the column vector associated with days off pattern j, and is consistent with the column vector used in (4).

3.3.6 Case III-C

Phase 1.

Same as in Case III-A

Phase 2.

Consistent with (3), there are ${}^7C_2 = 21$ days off patterns. The formulation is same as (9), except that j = 1,...21 and Aj is consistent with (3)

3.3.7 Case III-D

Phase 1:

Same as in Case III-B.

Phase 2

Consistent with (5), there are 3 days off patterns. The formulation is as follows:

$$\text{Minimize } C^F \sum_{j=1}^{3} x_j + C^P \sum_{j=1}^{3} y_j$$

$$\text{st.} \quad \text{(11a)} \quad \sum_{j=1}^{3} A_j x_j - F - Q = 0$$

$$\text{(11b)} \quad \sum_{j=1}^{3} A_j y_j - P + Q = 0$$

$$\text{(11c)} \quad 2F_1 + P_1 \geq 5(2\alpha^1 + \beta^1) \qquad \text{(11)}$$

$$2F_2 + P_2 \geq 2(2\alpha^2 + \beta^2)$$

$$\text{(11d)} \quad 5\gamma^1 \leq F_1 \leq 5\alpha^1$$

$$2\gamma^2 \leq F_2 \leq 2\alpha^2$$

(11e) $X \geq 0$, $Y \geq 0$, $F \geq 0$, $P \geq 0$, $Q \geq 0$ and integer vector;

where A_j is the column vector associated with days off pattern j, and is consistent with the column vector used in (5).

3.3.8 Case III-E

Case III-E provides an example of how the solution approach is complicated when full time shift is not an integer multiple of a part time shift. In the current formulation, we have allowed five-hour part time shifts.

Phase 1:

First, we need to establish the replacement relationships between FT and PT shifts.

FT to PT Replacement:

(i) 1 FT can be replaced by 2 PT .

PT to FT Replacement:

(ii) 1 PT can be replaced by 1 FT; 2 consecutive PT can be replaced by 2 FT.

(iii) 3 consecutive PT can be replaced by 2 FT.

For a16 hour workday, only 3 basic replacement relationships are found. (i) and (ii) are similar to that of Case III-A, so we only need to handle (iii).

One way to handle (iii) is to include in the formulation of phase 1 shift patterns of three PT shifts totaling 15 working hours. For a 16 hour work day, there are only 4 such shifts:

- Break, 3 consecutive PT
- 1 PT, Break, 2 consecutive PT
- 2 consecutive PT, Break, 1 PT
- 3 consecutive PT, Break.

The cost of these shifts would be slightly less than the cost of 3 PT employees, so that,

C^P = cost of one PT employee .

C^Z = cost of one employee with 15 hour shift pattern = $3C^P - \varepsilon$.

C^F = cost of one FT employee = $\frac{8}{3} C^P$.

Formulate the problem as in (2). Let the solution be as follows:

The number of FT employees equals αi

The number of PT employees equals βi

The number of employees with 15 hour shift patterns equals ξ_i

Phase 2

Formulate the connecting problem as in (9) except for a few modifications:

$$\text{Minimize } C^F \sum_{j=1}^{7} x_j + C^P \sum_{j=1}^{7} y_j$$

$$\text{St. (12a)} \quad \sum_{j=1}^{7} A_j x_j - F - Q - 2Z = 0$$

$$\text{(12b)} \quad \sum_{j=1}^{7} A_j y_j - P + Q + 3Z = 0 \qquad (12)$$

$$\text{(12c)} \quad 2F + P \geq 2\alpha + \beta + 3\xi$$

$$\text{(12d)} \quad \gamma \leq F \leq \alpha$$

$$\text{(12e)} \quad Z \leq \xi$$

$X \geq 0$, $Y \geq 0$, $F \geq 0$, $P \geq 0$, $Q \geq 0$, $Z \geq 0$, and are integer vectors and γ, β, α are all vectors corresponding to their indexed notations. Aj is the column vector associated with days off pattern j, and is consistent with the column vector used in (1).

Note that if $\xi_i = 0$ from phase 1, then $Z_i = 0$

where Z_i = number of groups of employees of the 15 hour shift pattern which are converted to FT employees.

3.3.9 2-Phase Algorithm Involving Multiple Contiguous Shift Patterns

An attempt is made here to devise a more general 2-phase algorithm based on Case (III-A) conditions, except that multiple contiguous shift patterns of arbitrary lengths are allowed instead of just full-time and part time shifts. The problem is difficult because a large number of possibilities exist for replacing one shift/combination of shifts by other shifts/combination of shifts which can satisfy demand for a span of time. When only full time and half time shifts are considered, two half time shifts can be replaced uniquely by a full time shift, or a full time shift can be replaced by two half time shifts. When multiple shifts of arbitrary lengths are involved, the number of replacement relationships is huge.

Some definitions of the terms to be used will be explained below.

Efficient Replacement Relationship:

When multiple shift patterns are allowed, many replacement relationships are possible. For example, any shift/combination of shifts can be replaced by another shift/combination of shifts in any instance when the span of working hours of the latter is longer than the former. However, in any replacement relationship, there exist at least one which performs the replacement in the most efficient way (i.e. with least excessive man-hours). Such replacement relationships are termed "efficient". In general, there can be several efficient replacement relationships in a given situation and it would be wasteful to include equivalent relationships in a mathematical model. Also, it would be redundant to include complex sets of relationships as individual alternatives in the model, when the model itself could construct the replacement relationships by combinations of simple relationships. We define these replacement relationships, which cannot be formed by a combination of other efficient replacements as being "basic".

Determining Efficient Replacement Relationships:

An approach for determining a set of basic efficient replacement relationships is given for a problem involving three different shifts. This approach can be generalized to handle any number of shifts but the process becomes tedious and requires an exhaustive computer search.

Given shift S_1 is l_1 hours and shift S_2 is l_2 hours, $l_1 > l_2$. Let r be the smallest integer greater than l_1/l_2. Then a basic efficient replacement relationship of S_1 by S_2 exists where r of S_2 can replace S_1; or $1(S_1) \Rightarrow r(S_2)$ where $\Rightarrow$ means can be replaced by.

- If $2(S_1) \Rightarrow 2r(S_2)$, this replacement relationship is not basic because it can be established by two cases of $1(S_1) \Rightarrow r(S_2)$.
- However, if $2(S_1) \Rightarrow p(S_2)$ where $p<2r$, then the replacement relationship is basic.
- If $3(S_1) \Rightarrow p(S_2) + r(S_2)$ or $3(S_1) \Rightarrow p(S_2), r(S_2)$ or $3(S_1) \Rightarrow (p + r)$ (S_2), this replacement relationship is not basic because it can be established through $2(S_1) \Rightarrow p(S_2)$ and $1(S_1) \Rightarrow r(S_2)$.
- However, if $3(S_1) \Rightarrow q(S_2)$ where $q<p+r$, then this replacement relationship is basic.
- Consider another shift $S_3 \Rightarrow 1(S_3) \Rightarrow 1(S_1)$.
  Since shift S_2 is shorter than S_1, $1(S_2) \Rightarrow 1(S_1)$. Let us assume that $2(S_2) \Rightarrow 1(S_1)$. If $1(S_2), 1(S_3) \Rightarrow 1(S_1)$, the replacement relationship is basic.

- If $2(S_2)$, $1(S_3) \Rightarrow 1(S_1)$, the replacement relationship is also basic.

- If $2(S_2)$, $2(S_3) \Rightarrow 2(S_1)$, that replacement relationship is not basic because the replacement could be established through 2 cases of $2(S_2)$ and $1(S_3)$, or $1(S_2)$, $1(S_3)$ and $1(S_2)$.

- If $1(S_2)$, $2(S_3) \Rightarrow 1(S_1)$, the replacement relationship is basic.

- If $1(S_2)$, $2(S_3) \Rightarrow 2(S_1)$, that replacement relationship is not basic.

To summarize, if an efficient replacement relationship is such that it cannot be established through a combination of simpler replacement relationships, then it is basic and need not be represented in the mathematical model.

Joint Shifts

A joint shift is a combination of shifts. Examples of joint shifts are $2(S_1)$ or $1(S_1)$, $1(S_2)$ or $3(S_3)$, $2(S_1)$. The shifts forming the joing shift are always non-overlapping

A statement of the algorithm is given below:

Step 1. Establish all basic efficient replacement relationships between shifts/joint shifts. If a basic replacement relationship involves replacing joint shifts, the latter have to appear as column vectors in Phase 1.

Step 2. Phase 1 costs are ordered and longer shifts/joing shifts are costed proportionately higher.

Step 3. Convert all Phase 1 solutions in terms of the shortest shift type.

Step 4. Except for the shortest shift, Phase 1 solutions provide the upper bounds to all other shifts/joint shifts, to be used in Phase 2.

Step 5. In Phase 2, all basic replacement relationships between shifts/joint shifts (except for direct replacement of basic shifts by the shortest shift), will have to be represented as variable in the formulation, thus, allowing shift/joint shifts to move from one type to another.

Let us apply the preceding algorithm to a simple problem involving 3 different shifts - S_1, an 8 hour shift, S_2, a 5 hour shift, and S_3, a 3 hour shift, and a working day of 10 hours.

Step 1:

$S_1 \Rightarrow S_3$	Ratio	Phase 2 Variables
1(8) ⇒ 3(3)	3:1 *(Basic replacement)	Not applicable because all S_1 shifts are converted to S_3 shift equivalents.
$S_2 \Rightarrow S_3$		
1(5) ⇒ 2(3)	2:1 *	Not applicable because all S_2 shifts are converted to S_3 shift equivalents.
2(5) ⇒ 4(3)	2:1	
$S_1 \Rightarrow S_2$		
1(8) ⇒ 2(5)	2:1 *	P^1
$S_1 \Rightarrow S_1$		
1(5) ⇒ 1(8)	1:1 *	P^2

$\underline{S_3 \Rightarrow S_1}$

$1(3) \Rightarrow 1(8)$	1:1 *		P^3
$2(3) \Rightarrow 1(8)$	1:2 *	joint shift (J_1)	P^4
$3(3) \Rightarrow 2(8)$	2:3 = (1:1) + (1:2)		

$\underline{S_3 \Rightarrow S_2}$

$1(3) \Rightarrow 1(5)$	1:1 *		P^5
$2(3) \Rightarrow 2(5)$	1:1		
$3(3) \Rightarrow 2(5)$	2:3 *	joint shift (J_2)	P^6

$\underline{(S_2\ ,\ S_3) \Rightarrow S_1}$

$1(5),\ 1(3) \Rightarrow 1(8)$	1:1,1 *	joint shift (J_3)	P^7
$2(5),\ 1(3) \Rightarrow 2(8)$	2:2,1 = (1:1,1)+(1:1,0)		

$\underline{S_1 \Rightarrow (S_2,\ S_3)}$

$1(8) \Rightarrow 1(5),\ 1(3)$	1,1:1 *		P^8

Joint shift (J_1) has to be reoresented as column vector in Phase 1.

e.q. S_3, S_3, 4B   Where B = break for an hour

S_3, 2B, S_3, 2B

and so on.

Similarly for joint shift (J_2)

e.q. S_3, S_3, S_3, B

B, S_3, S_3, S_3

S_3, B, S_3, B

Joint shift (J_3) and (J_4), each has a total working length of 8 hours which is S_1 itself. Therefore they do not have to be considered.

Step 2: Essentially, we have 3 basic shifts (S_1, S_2, S_3) and 2 joint shifts (J_1, J_2). Order them in descending order: J_2, S_1, J_3, S_2, S_3. The longer shifts are cost proportionately higher. Phase 1 problem is shown below.

$$\begin{aligned} \text{Min.} \quad & C \; (J_2, S_1, J_3, S_2, S_3)^T \\ \text{st} \quad & A \; (J_2, S_1, J_3, S_2, S_3)^T \geq R \\ & (J_2, S_1, J_3, S_2, S_3)^T \geq 0 \end{aligned} \qquad (15)$$

Step 3: Phase 1 solution would consist of the following, converted to the shortest shifts S_3,

The number of S_1 shift	=	$\alpha^1 \Rightarrow 3\alpha^1 \quad S_3$
" " " S_2 " "	=	$\alpha^2 \Rightarrow 2\alpha^2 \quad S_3$
" " " S_3 " "	=	$\alpha 3$
" " " J_1 " "	=	$\alpha^4 \Rightarrow 2\alpha^4 \quad S_3$
" " " J_2 " "	=	$\alpha^5 \Rightarrow 3\alpha^5 \quad S_3$

Step 4:

α^4	is upper bound for	P^4
α^5	" "" " "" "	P^6
α^1	" "" " "" "	$F^1 + P^7$
α^2	" "" " "" "	F^2

Step 5: The Phase 2 problem becomes:

$$\text{Min} \quad C(x, y, z)^T$$

$$\text{st} \quad (16a) \quad \Sigma A_j x_j - F^1 + P^1 - P^2 - P^3 - P^4 \qquad - P^7 + P^8 = 0$$

$$(16b) \quad \Sigma A_j y_j - F_2 - 2P^1 + P^2 \qquad - P^5 - 2P^6 + P^7 - P^8 = 0$$

(16c) $\Sigma A_j z_j - F^3 \qquad + P^3 + 2P^4 + P^5 + 3P^6 + P^7 - P^8 = 0$

(16d) $3F^1 + 2F^2 + F^3 \geq 3\alpha^1 + 2\alpha^2 + \alpha^3 + 2\alpha^4 + 3\alpha^5$

(16e) $F^1 + P^7 \leq \alpha^1 \qquad P^4 \leq \alpha^4$

$F^2 \leq \alpha^2 \qquad P^6 \leq \alpha^5$

(16f) $x \geq 0$, $y \geq 0$, $z \geq 0$, $F^i \geq 0$, $P^i \geq 0$ and integer vectors, α^i are integer vectors for all i.

3.4 Problem Type IV

3.4.1 Extension to 24 hours Work Week

We have considered the scheduling of manpower to satisfy hourly manpower requirements subject to days off patterns for a workday which is no longer than 16 hours. In cases where the working day lasts to 24 hours, our 2-phase algorithm as it stands presently, would provide a good schedule, although not necessarily optimal, because shifts that work throughout the midnight hour are not considered in the context of our model. Consider now the general problem of scheduling full time and part time employees such that periods off (instead of days off) are considered. A full time employee would work for 8 hours, break (periods off) for 12 hours, work the next 8 hours, break for 15 hours, work the next 8 hours and so on. The idea is to have a full time employee work 5 periods of 8 hours having "reasonable" breaks in between. Similarly, a part time employee would work periods of 4 hours subject to having "reasonable" breaks in between.

Let us consider an hour by hour manpower requirement for the whole week cycle. There are a total of 24x7 = 168 hours per week. The problem can be formulated as an integer program:

$$\text{Min} \quad \Sigma \; c_j x_j$$

$$\sum_j A_j x_j \geq R \qquad (17)$$

$x_j \geq 0$ and integer and $A_j = (a_{1j}, a_{2j} \ldots\ldots a_{168j})^T$

$R = (R_1, R_2 \ldots\ldots R_{168})^T$

All notations are similar to those of (2).

As one can see, the problem is a large one, requiring at least 168 constraints, and the number of variables can run to millions, since the number of shift/periods off patterns is very huge.

Let us try to apply the 2-phase algorithm developed earlier to the present problem based on Case III-A conditions, except for condition (3), where the days off now become "reasonable" periods off.

Phase 1

Solve (17) where the column vectors A_j's consist of either 4 hour consecutive or 8 hour consecutive shifts

Cost FT shifts C^F and PT shifts C^P such that $C^F = 2C^P - \varepsilon$

Since we have to consider shift patterns that work through the 168th hour and back to the 1st hour, the problem consists of contiguous and split shift patterns. From (6), we have learned that problem with the above structure can be solved by the Bender's algorithm if necessary.

Assume now that we obtain the following solution·

α_f = number of FT in shift pattern f. $f = 1,2, \ldots F$ $F \leq 168$

β_g = number of PT in shift pattern g $g = 1,2, \ldots G$ $G \leq 168$

Phase 2

Phase 2 problem will attempt to create sets of 5 FT shifts and 5 PT shifts with "reasonable" breaks. Given α_f and β_g, it is possible to enumerate all compatible sets, although the number of combinations might be huge. Granted that, it is also possible to enumerate all possible replacement relationships between part time and full time shifts (e.g. PT shifts $\{A_j \mid a_{ij}$,

$a_{(i+1)j}, \cdots a_{(i+3)j} = 1\}$ or $\{A_j \mid a_{(i+1)j}, a_{(i+2)j}, \ldots, a_{(i+4)j} = 1\}$ can be replaced by FT shift $\{A_j \mid a_{ij}, a_{(i+1)j}, \cdots a_{(i+7)j} = 1\}$

Now any FT shift pattern can be broken up into 2 equivalent part time patterns, e.g. α_f FT = α_f of gth PT pattern and α_f of $(g + 4)$th PT pattern. The total number of PT equivalents working PT shift pattern g = $\beta_g + \alpha_f + \alpha_f^1$ where if f_t = hour shift pattern f starts, then shift pattern f^1 starts at $(f_t - 4)$ hour.

Example:

For simplicity of explanation, let us assume that every hour (all 168 hours) appear in the solution of phase 1, i.e. we have α_f, f = 1,2, ..., 168, and β_g, g = 1, 2, ..., 168. Total PT pattern i equivalents = $(\beta_i + \alpha_i + \alpha_{i-4})$ where i = 1,2, ..., 164, 165 (-3), 166 (-2), 167 (-1), 168 (0). Any PT shift pattern i can be replaced by FT shift patterns i, i-1, i-2, i-3, i-4, remembering the cyclic nature of the shift pattern. Q^k, k=1, ..., 5 is used to represent the replacement of PT shift by FT shift. In compact form, then, the formulation can be written, similar to (11):

$$\text{Min} \quad C^F \sum_i x_i + C^P \sum_i y_i$$

$$\text{st (18a)} \quad \sum_i A^F_i X_i - F - \sum_{k=1}^{5} Q^K = 0$$

$$\text{(18b)} \quad \sum_j A^P_j y_j - P + \sum_{k=1}^{5} Q^K = 0 \qquad (18)$$

$$\text{(18c)} \quad 2F + P \geq \beta + \alpha + \alpha_{(I-4)}$$

(18d) $\gamma \leq F \leq \alpha$

where:

$$\alpha_{I-4} = (\alpha_{165}, \alpha_{166}, \ldots, \alpha_{168}, \alpha_1, \alpha_2 \ldots, \alpha_{164})^T$$

(18e) $x \geq 0$, $y \geq 0$, $F \geq 0$, $P \geq 0$, $Q^K \geq 0$, $\alpha \geq 0$, $\alpha_{I-4} \geq 0$, $\beta \geq 0$, $\gamma \geq 0$, are integer vectors. $A^F{}_j$ is the column vector associated with a compatible set of FT shifts and $A^P{}_j$ are associated with a compatible set of PT shifts.

(18) can become quite huge, depending on how many of the shift patterns appear in the solution of Phase 1, and the number of sets of compatible shift patterns.

It is interesting to note that for the Phase 1 problem if the hourly requirements over the day are similar for every day of the week, the same results could be obtained by breaking the problem up into 7 equal problems since the optimal solution for one day would be the optimal for another day. Consider the following problem involving 3 periods a day and shifts of 2 periods in length.

Days	Per-iods	shift patterns for a week cycle (ordered from 1, ... 21, 1, ...)																							Require-ments per periods
M	1	1	1																						R_1
	2		1	1																					R_2
	3			1	1																				R_3
Tu	1				1	1																			R_1
	2					1	1																		R_2
	3						1	1																	R_3
W	1							1	1																R_1
	2								1	1															R_2
	3									1	1														R_3
Th	1										1	1													R_1
	2											1	1												R_2
	3												1	1											R_3
F	1													1	1										R_1
	2														1	1									R_2
	3															1	1								R_3
Sa	1																1	1							R_1
	2																	1	1						R_2
	3																		1	1					R_3
Su	1																			1	1				R_1
	2																				1	1			R_2
	3																					1	1		R_3
M	1																						1	1	R_1

(19)

The manpower requirements are similar for each day, i.e. requirements in period 1, Monday is the same as requirements in period 1, Tuesday and so forth. Because of the cyclic nature of the daily requirements,

shift (1) = shift (4) shift (7) = shift (10) . . . = shift (22)

and shift (2) = shift (5) = shift (8) = shift (11) . . . = shift (20)

and shift (3) = shift (6) = shift (9) = shift (12) . . . = shift (21)

We shall break up the problem into 7 equal problems of the following structure:

Period	Shifts			Requirements
1	1	1		R_1
2		1	1	R_2
3	1		1	R_3

shift (2)

shift (1)=shift (4)   shift (3)

Hence instead of solving a 7 day problem, we need only solve a 1 day problem.

Heuristics

Below is an attempt to solve a more general manpower scheduling type 4 problem where multiple shifts including split shifts are involved: using a heuristic approach. The heuristic approach involves 2 phases as before. Phase 1 solves (17) where all shift patterns to be considered appear as column vectors. Since (17) is quite huge, rather than solving it with an integer program code, divide the problem into several smaller problems with equal number of constraints. Let us divide (17) into 7 equal problems. There will be some variables common to sub-problems (i) and (i + 1). Call these common variables. Now solve sub-problem (1) using an integer program code. If some common variables appear in the optimal solution, they should be subtracted from the requirements vector of sub-problem (2) accordingly. Now solve sub-problem (2) and so on until sub-problem (7). Sub-problem (7) will have common variables of sub-problem (1) and (6) subtracted from the requirements vector. We have obtained now as our solution, sets of shift patterns S_1, S_2, S_3, .., S_n (in descending order of shift lengths).

Phase 2 looks at the Phase 1 solution and tries to create sets of 5 S_1 shifts with reasonable breaks. If there are leftovers, break them up to shorter shifts, adding as few extra man-hours as possible. (e.g. breaking 8 hour shift to a 5 and 3 hour shift is better than breaking an 8 hour shift to three 3 hour shifts or two 5 hour shifts.) Create sets of 5 S_2 shifts. Again if there are leftovers, break them up to shorter shifts, and so on until S_n shifts. If there are leftovers in S_n shifts, 3 things can be done: (a) allow over time for employees already scheduled, (b) allow some S_n shift employees to work less than 5 S_n shifts a week, (c) add in more S_n shifts to create a new set.

Chapter VI

Results and Discussions

4.1 Problem Type I (Days off Scheduling)

Problems of this type are generally small, usually not more than 7 constraints. For problems with special structure, the number of variables and the number of constraints can be further reduced as can be seen in (4) and (5). "Good" or optimal solutions to problems of this type can usually be arrived at by enumeration within a short time. Consequently problem type I is of interest primarily because it comprises of subunit in more complex formulations.

In all the formulations of problem type I, we have confined ourselves to employees working a 5-day week. It is conceivable to have employees working 6-day week (overtime) or 4-day week (part-time), etc. The formulation can easily be modified by adding the required days-off patterns.

4.2 Problem Type II (Shift Scheduling)

In most instances type II problems contain up to a hundred constraints and hundreds of variables. It was shown that the use of Benders Partitioning Method to solve the problem, is a viable alternative to a branch and branch Integer Program algorithm. The number of Benders cut for all problems solved (not withstanding the size of the problem) was always less than 7. This means that the largest integer program used has less than 7 constraints.

Exhibit 1 shows some of the results obtained for a few randomly selected problems. The branch and bound code used was based on Linear

Program with DKW/Tomlin's penalties [9, 31] and the transshipment code used was based on Ford and Fulkerson's primal-dual algorithm [10].

Exhibit 1

Type II Problems

Sample Problem Number	Number of Working Hours	Number of Shift Patterns	Number of Benders Cut	Computer Running Times on CDC 6600
1	10	14	7	7 sec.
2	12	24	6	8 sec.
3	42	84	7	70 sec.
4	84	168	6	317 sec.

The slow running times for larger problems 3 and 4 were attributed to the fact that the transshipment code used was far more inefficient than expected; for example, the fourth problem of Exhibit 1 required approximately 50 seconds to solve each 84-nodes transshipment problem. It is reported in the literature [14] that there are codes which can solve a 200-node transshipment problem in 1.3 seconds on a CDC 6600. Consequently, one could expect to solve the fourth example problem in ten seconds.

It is a difficult task to prove that Benders Partitioning Method is a better algorithm than the branch and bound algorithm for solving problem Type II. One has to determing whether there are savings in storage and/or in computational times and both depend on the particular codes being used. Much more work has to be put in before any meaningful comparison can be made.

4.3 Problem Type III (Shift and Days Off Scheduling)

Randomly selected Type III problems were run using the 2-phase algorithm based on Case (III-A) conditions including split shifts. Exhibit 2 shows some of the results. The algorithm was shown to be extremely efficient. The maximum running times for the few selected problems do not exceed 14 seconds on a CDC 6600. The integer program subroutine used was the same one mentioned above.

Exhibit 2

Type III Problems

Sample Problem Numbers	Number of Working Hours for Day Number							Computer Running Times on CDC 6600
	1	2	3	4	5	6	7	
1	10	10	10	10	10	12	12	11.7 sec.
2	12	12	12	12	12	12	12	13.1 sec.
3	9	10	13	10	10	12	12	12.2 sec.
4	16	16	10	10	10	12	12	12.4 sec.

When part time shift is not a 4 hour shift, the 2-phase algorithm can be modified to handle the new problem. However, because the replacement relationship between the full-time and part-time shifts is not as simple as the 8 hour PT and 4 hour PT problem before, the problem formulation sizes in both Phase 1 and Phase 2 are increased appreciably as shown for Case III-E in Section 3.3.8.

When multiple shifts are involved, the 2-phase algorithm becomes even more complicated. Replacement relationships between shifts/joint shifts become more complex. For a problem involving 3 different shift patterns (Section 3.3.9) Phase 1 problem could have as many as a hundred variables, although the number of constraints remains the same. However, in Phase 2, the number of variables is as many as 98, and the number of constraints is 56 for which 21 are upper/lower bounds on the variables. It appears from the problem sizes, that the 2-phase algorithm is quite efficient even for a problem involving 3 different shift patterns.

It is noted here that for problems involving simple replacement relationship (e.g. 8 hour full time and 4 hour part time shifts), the 2-phase algorithm is extremely efficient. However, as the number of different shift patterns increases, the replacement relationships become more complex, and hence difficult to establish. Also the Phase 1 and 2 problems become very large and hence inefficient to solve.

It can be shown that Formulation II (as presented in Section 3.3.1.2. for full time and part time shifts) can easily be extended to multiple shifts. The advantage in using Formulation II is that, it does not require the process of establishing replacement relationships. Also the "A" matrix in Formulation II is highly structured. This dissertation has not been successful in exploiting the structure of the "A" matrix. Suffice to say, this should be interesting work for future research.

4.4 Problem Type IV (Shift and Periods off Scheduling)

Since shifts that work through the midnight hour have to be considered, it is not possible to split up the week into seven days as in Problem Type III. When the 2-phase algorithm was applied to the type IV problem, it is necessary to consider the whole week in one stage. The phase 1 problem has 168 constraints and 336 variables. Benders Partitioning Algorithm as presented in Section 3.2 could help to solve the problem. Granted that, the Phase 2 problem required generating compatible shift patterns from Phase 1 results, and the number of constraints depend on the number of shift patterns appearing in Phase 1. Hence the size of Phase 2 problem can only be guessed. An estimate would be as many as 200 constraints and as many variables.

In general, where multiple shifts are involved, it might be wiser to resort to the heuristic algorithm as presented in Section 3.4. Although the heuristic algorithm is not tested, it is justified to expect that the algorithm would give efficient schedules since it involves using many of the ideas used in the 2-phase algorithms.

4.5 Automated Manpower Scheduling System (AMSS)

An Automated Manpower Scheduling System (AMSS) was designed for the U.S. Army Food Service Operations. AMSS essentially consists of two computer programs SCHED and MANPOW written in FORTRAN·

(1) SCHED develops the optimal employee schedule for a day, given that the minimum manpower requirements fluctuate from hour to hour. It is a typical type II problem. The algorithm involves using the Integer Program Code as a subroutine. (Benders algorithm was not used).

(2) MANPOW develops the optimal employee schedule for a week, given that the minimum manpower requirements fluctuates from hour to hour and day to day. It is a typical type III problem based on Case III-A conditions except that split shifts as well as undesirable contiguous shifts can be specified by the user.

Details of both programs can be obtained from the AMSS User's Manual, a copy of which can be found in Appendix I.

Chapter 5

Conclusion

Four different types of manpower scheduling problems in Service Operations were discussed, each having its own unique structure; some of them are more amenable to algorithmic exploitation than others, alluding to the need for using different algorithms for different problem type. At least five algorithms were developed or proposed - Integer program, transshipment algorithm, Benders Partitioning method, 2-phase algorithm and the heuristic algorithm.

Although some of the proposed algorithms have been tested, and found to be efficient, the real test of an algorithm only comes when the results are implemented in a real situation.

Plans are now underway to implement the Automated Manpower Scheduling System (AMSS) in selected Army Camps by the U.S. Army Natick Laboratories. Their findings would reflect to some extent the usefulness of this dissertation.

Data which are assumed to be known for a scheduling problem, like the hourly/daily minimum manpower requirements could be difficult to estimate. Although demand for services is assumed to be an uncontrollable factor, this is not always the case in real situations because demand for services can be smoothed or even changed by offering different kinds of incentives, e.g., special rates for telephone calls in the night. Much more has to be studied concerning the degree of smoothing or change in service demand for any typical service operation. In fact, a research project directly addressing the data collection problem is being conducted at the Industrial Engineering and Operations Research Department at the University of Massachusetts.

Innovations in terms of flexibility in working hours,shift and days off patterns and work cycles, all contribute to the need for improving scheduling models so as to develop better schedules. The future should see some departure from the traditional 5-day week, 8 hour day schedule, moving, for example, to 4 consecutive 10-hour days or 6 consecutive work days every 10 days. This would result in more complex scheduling problems, necessitating even more so the use of complex algorithms and computers.

REFERENCES

[1] Abernatty, W.J., et al, "A Three Stage Manpower Planning and Scheduling Model - A Service Sector Example", Operations Research, Vol. 21, No. 3, May-June 1973.

[2] Ahuja, H. and Sheppard, R., "Computerized Nurse Scheduling", Industrial Engineering Journal, October 1975, Vol. 7, No. 10.

[3] Arabeyre, J.P., et al, "The Airline Crew Scheduling Problem, A Survey", Transportation Science, Vol. 3, No. 2, May 1969.

[4] Armatas, J.P. and Lundbert, D.E., "The Management of People in Hotels, Restaurants and Clubs", Wm. C. Brown Co., Dubuque, Iowa.

[5] Baker, K.R., "Scheduling a Full Time Work Force to Meet Cyclic Staffing Requirements", Management Science, Aug. 1974.

[6] Benders, F.R., "Partitioning Procedures for Solving Mixed Variables Programming Problems, Numerische Mathematik 4, 1962.

[7] Charnes, A., Cooper, W.W., and Niehaus, R.J., "Studies in Manpower Planning," U.S. Navy Office of Civilian Manpower Management, Washington, D.C., July, 1972.

[8] Conway, R.W., Maxwell, W.L., and Miller, L.W., "Theory of Scheduling", Addison Wesley, 1969.

[9] Davis, R.E., Kendrick, D.A., and Weitzman, J., "A Branch and Bound Algorithm for 0-1 Mixed Integer Programming Problems", Development Economic Report No. 69, Center for International Affairs, Harvard University.

[10] Ford, L.R., Jr., and Fulkerson, D.R., "Flows in Networks," Princeton University Press., 1962.

[11] Geoffrion, A.M. and Graves, G.W., "Multi-Commodity Distribution System Design by Bender's Decomposition", Management Science, Jan. 1974.

[12] Gere, W.S., Jr., "Heuristics in Job Shop Scheduling", Management Science, Vol. 13, No. 3, 1966.

[13] Giglio, R.J., and Noonan, F., "A Mathematical Programming Model for Long Range Planning of Electric Power Generation", Technical Paper presented at the ORSA/TIMS Puerto Rico Meeting, Jan. 1974.

[14] Glover, F., Karney, D. and Klingman, D., "Implementation and Computational Comparisons of primal, dual and primal-dual computer codes for minimum cost network flow problems", Center for Cybernetic Studies, The University of Texas, Austin.

[15] Green, J.H., "Operations Planning and Control", Richard D. Irwin, Inc., Homewood, Illinois, 1967.

[16] Hannan, E. and Giglio, R.J., "An algorithm for Scheduling Physicians for Outpatient Clinic", Technical Paper presented at the 42nd National meeting of ORSA/TIMS and AIIE, Atlantic City, NJ, Nov. 1972.

[17] Healy, W.C., "Shift Scheduling Made Easy", Factory, October 1957.

[18] Jessop, W., Editor, "Manpower Planning: Operational Research and Personnel Research", New York, American Elsevier Publishing Co., 1966.

[19] Lasdon, L.S., "Optimization Theory for Large Systems", Macmillan series in Operations Research, 1970.

[20] Luce, W.J., "A Shift Scheduling Algorithm", Technical paper presented at the 44th National Meeting of ORSA, San Diego, California, Nov. 1973.

[21] Luce, W.J., "Employee Assignment System", Technical Paper presented at Joint National Meeting of ORSA/Tims, Boston, Mass., April 1974.

[22] Mapstone, B.E., and Thamara, T., "Vacation Scheduling" Industrial Engineering Journal, May, 1976, Vol. 18, No. 5.

[23] Monroe, G., "Scheduling Manpower for Service Operations", Industrial Engineering, August 1970.

[24] Muth, J.E., and Thompson, G.L., "Industrial Scheduling", Prentice-Hall, 1963.

[25] O'Brien, J.J., "Scheduling Handbook", New York, Mc-Graw-Hill, 1969.

[26] Rothstein, J., "Scheduling Manpower by Mathematical Programming," Industrial Engineering, April 1972.

[27] Segal, M., "The Operator Scheduling Problem: A Network Flow Approach", Journal of ORSA, July 1974.

[28] Smith, R.S., "Two-Phase Employee Scheduling Algorithm for Operations Having Variable Manpower Requirements with Applications Involving Single and Composite Planning Cycles", Doctoral Dissertation, University of Massachusetts, Feb. 1975.

[29] Smith, R.S., et al, "A Systems Evaluation of Army Garrison Feeding at Fort Lewis, Washington, Tech. Report No. 72-37, OR/SA, Jan. 1972. U.S. Army Natick Labs.

[30] Tibrawala, R., et al, "Optimal Scheduling of Two Consecutive Idle Periods", Management Science, Vol. 19, No. 1, Sept. 1972.

[31] Tomlin, J.A., "Branch and Bound Methods for Integer and Non-Convex Programming", in Integer and Non-Linear Programming. Ed. J. Abadie, Amsterdam, North Holland Publishing Co., 1970.